



# Fundamentos de Big Data

## Lección 4: Introducción a Data Mining

# ÍNDICE

<b>Lección 4. – Introducción a Data Mining.....</b>	<b>2</b>
<b>Presentación y objetivos .....</b>	<b>2</b>
<b>1. Nos unimos a la Competición de Titanic .....</b>	<b>3</b>
<b>2. Explicación del Titanic Dataset .....</b>	<b>5</b>
<b>3. Primeros pasos con Titanic Dataset.....</b>	<b>9</b>
<b>4. Selección de información en DataFrames .....</b>	<b>15</b>
<b>5. Creando “nuevos” DataFrames .....</b>	<b>21</b>
<b>6. Obtención de Información de los Gráficos.....</b>	<b>23</b>
<b>7. Puntos clave .....</b>	<b>37</b>

# Lección 4. – Introducción a Data Mining

## PRESENTACIÓN Y OBJETIVOS

Llegados a este punto deberíamos saber unas pocas cosas acerca de las gráficas que pueden emplearse según el momento. No obstante, existen más tipos de gráficas.

Si recordamos, en la Asignatura de Creación de Aplicaciones Python estuvimos viendo unos pequeños conocimientos en Machine Learning.

Una gran forma de poner en práctica lo aprendido hasta el momento y ver unas pocas cosas más es utilizar el Titanic Dataset para profundizar en ambos puntos a un mismo tiempo.

En esta lección haremos cosas muy útiles para cualquier estudiante de un programa académico relacionado con Inteligencia Artificial.



### *Objetivos*

- Conocer el Titanic Dataset
- Conocer algunas cosas más de Machine Learning
- Aprender algo de Data Mining

## 1. NOS UNIMOS A LA COMPETICIÓN DE TITANIC

Ahora lo que debes hacer es “Join Competition” y aceptar las reglas.

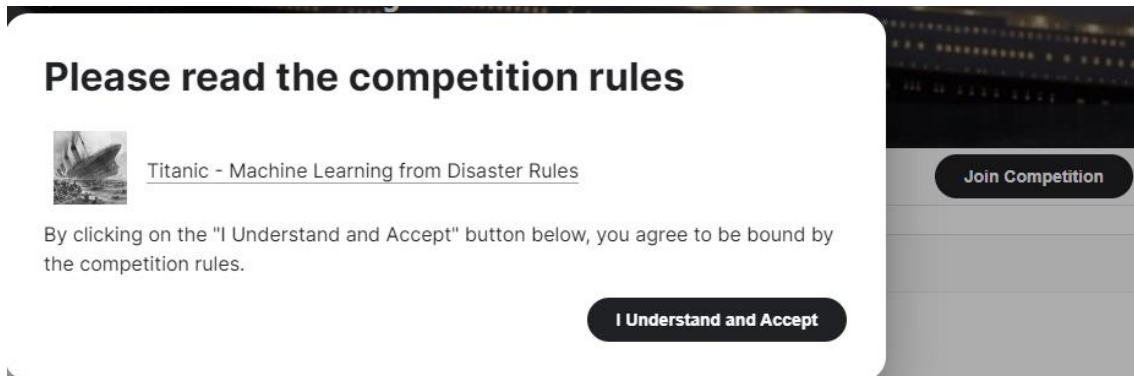


Figura 1.1: Titanic dataset en Kaggle (parte 1)

Tan simple como eso.

Por cierto, con Kaggle se podrían ganar medallas que de alguna manera demuestran nuestras habilidades.

Podríamos ver las “medallas” que ganamos en “Your Profile”.

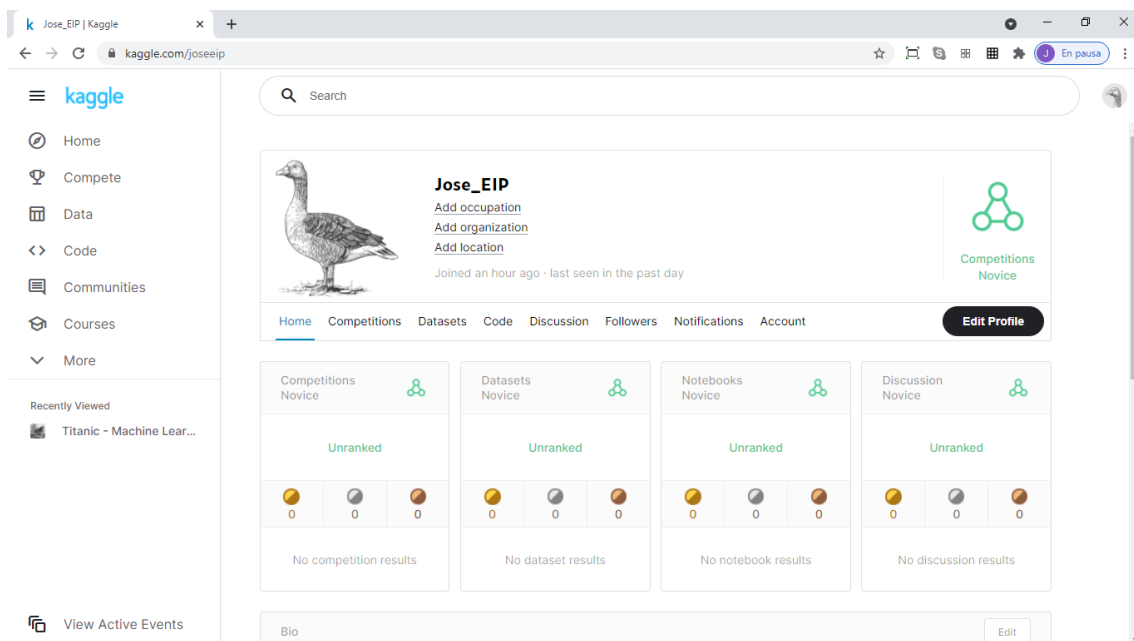


Figura 1.2: Titanic dataset en Kaggle (parte 2)

Ahora si nos vamos a “Compete” veremos que estamos registrados en la competición, la cual sirve como aprendizaje. (Knowledge)

Lo vemos en “Your Competitions” (Tus competiciones).

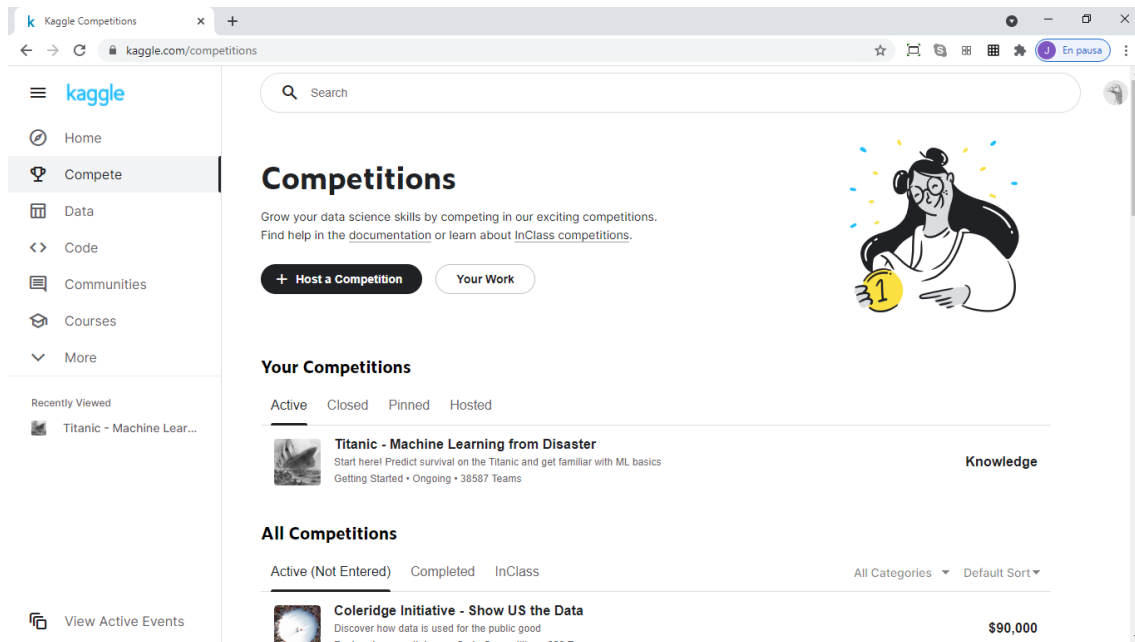


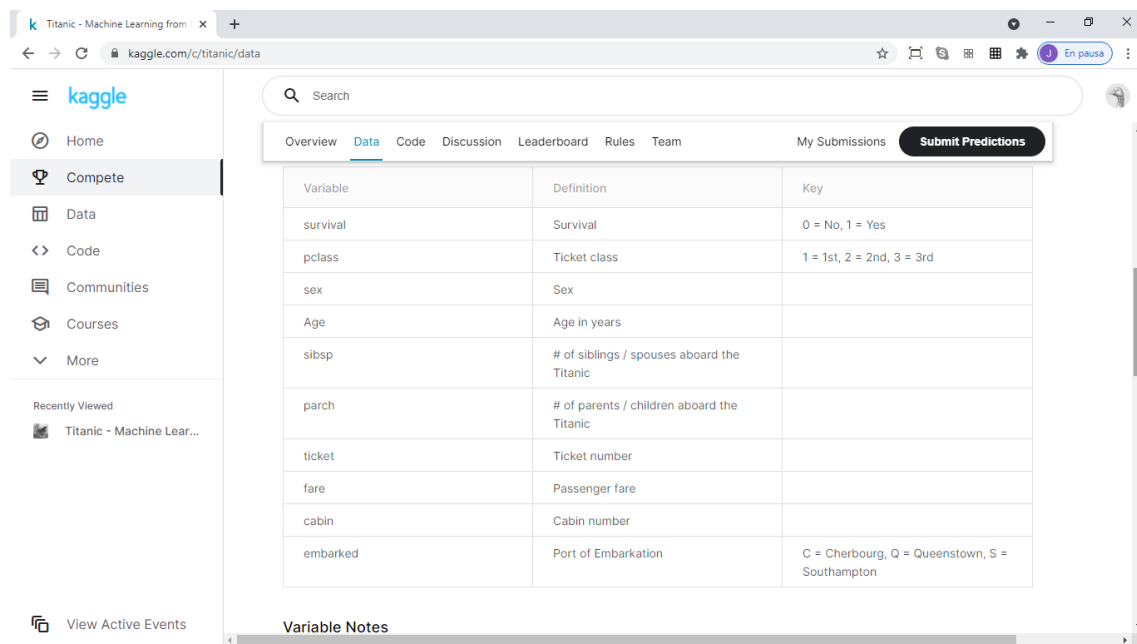
Figura 1.3: Titanic dataset en Kaggle (parte 3)

Existen más cosas que se podrían explicar, no obstante, lo más importante para esta lección es el propio dataset.

## 2. EXPLICACIÓN DEL TITANIC DATASET

Lo primero es entender qué tenemos.

Para ello nos vamos aquí:



The screenshot shows the Kaggle website interface for the Titanic dataset. The left sidebar contains navigation links: Home, Compete, Data, Code, Communities, Courses, and More. The main content area has tabs for Overview, Data (selected), Code, Discussion, Leaderboard, Rules, and Team. Below the tabs is a table with three columns: Variable, Definition, and Key. The table lists variables such as survival, pclass, sex, Age, sibsp, parch, ticket, fare, cabin, and embarked, along with their definitions and keys. A 'Submit Predictions' button is visible in the top right corner.

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Figura 2.1: Explicación del Titanic dataset (parte 1)

Que nos va a proporcionar la información.

Como en la mayoría de ocasiones en Software, la información se encuentra en idioma Inglés.

Se puede hacer uso de traductores, si desconocemos lo que dice, en este caso lo explicaremos en español también, para ahorrar ese trabajo.

Si ampliamos la información dice lo siguiente.

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Figura 2.2: Explicación del Titanic dataset (parte 2)

Explicación de las variables en idioma español:

- **Survival = Supervivencia (si o no)**  
Es decir, una variable booleana, 1 ó 0.  
Por ello un problema de “Clasificación binaria”
- **Pclass = Clase**  
1ª clase supuestamente la de aquellas familias más adineradas  
2ª clase: nivel intermedio  
3ª clase: la de aquellas familias más humildes.
- **Sex = Sexo**  
Masculino o femenino. (1 ó 0)
- **Sibsp = Número de hermanos/as o esposos/as a bordo del Titanic.**  
Es un número obviamente.
- **Parch = Número de padres/hijos a bordo del titanic**  
Un número obviamente

- **Ticket = Número del billete**  
Veremos después si tiene o no relevancia
- **Fare = Tarifa, precio del billete**  
¿ A mayor precio mayor capacidad de supervivencia ? Lo veremos..
- **Cabin = Número de Cabina**  
Veremos si tiene relevancia o no.
- **Embarked = Lugar desde donde se había embarcado**  
Son 3 lugares que vienen indicados en la imagen.

Una vez nos hacemos a la idea de la información que tenemos

Nos vamos a descargar los datos en nuestro PC.

A continuación explicaremos que significa cada cosa.

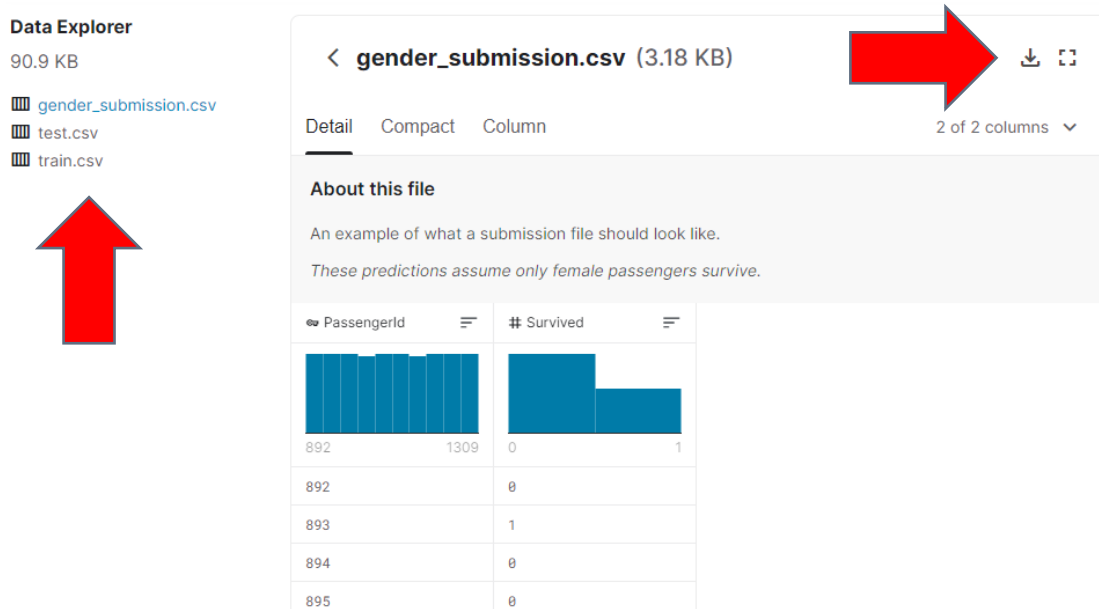


Figura 2.3: Explicación del Titanic dataset (parte 3)



Tenemos:

- **train.csv**

Al importar ese .csv tendríamos nuestro dataframe, que es lo que conocíamos normalmente como “df”.

Que no despiste lo de train.csv y test.csv.

Así pues, dentro de train.csv, tendremos entrenamiento y prueba.

train y test (X\_train, X\_test, y\_train, y\_test).

- **test.csv**

Es la información sobre la cual vamos a predecir.

- **gender\_submission.csv**

Es el .csv que tendremos que proporcionar para que automáticamente nos digan el número de aciertos y fallos.

Posteriormente veremos cómo se hace eso.

- ✓ Comentario 1:

Los modelos de predicción de este tipo en los cuales tratamos de predecir una salida en función de unos datos de entrada se llama “Aprendizaje Supervisado”.

- ✓ Comentario 2:

En este caso tenemos un ejemplo de “Clasificación binaria”

En el caso de Iris Dataset había 3 salidas, con lo cual, “Clasificación Multivariable”.

- ✓ Comentario 3:

Otro posible tipo en aprendizaje supervisado, sería la regresión.

Ejemplo: Producir el precio de los alquileres en una zona de la ciudad en base al precio pasado.

- ✓ Comentario 4:

Respecto “Aprendizaje Supervisado” y “No Supervisado” se aprenderá mucho mejor en las Asignaturas de Machine Learning, que se encuentran posterior a las actuales de Big Data.

### 3. PRIMEROS PASOS CON TITANIC DATASET

#### Importamos nuestras principales dependencias

```
In [1]: # principales dependencias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: # Importamos algunos algoritmos de clasificación
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

In [3]: # una posible forma para evaluar nuestro modelo
from sklearn.metrics import accuracy_score
```

Figura 3.1: Primeros pasos con Titanic Dataset (parte 1)

#### Tratamos de entender el problema

```
In [4]: # https://www.kaggle.com/c/titanic
# ya explicado en el manual
```

#### Obtención de datos

```
In [5]: # C:\Users\Manut\Desktop\apuntes_big_data_1\TEMA 4\data
df = pd.read_csv("C:/Users/Manut/Desktop/apuntes_big_data_1/TEMA 4/data/train.csv")
df.head()
```

```
Out[5]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Figura 3.2: Primeros pasos con Titanic Dataset (parte 2)

### Borro la columna PassengerID

```
In [6]: df = df.drop('PassengerId',axis=1)
```

Figura 3.3: Primeros pasos con Titanic Dataset (parte 3)

### Exploratory Data Analysis (EDA)

```
In [7]: df.head()
```

```
Out[7]:
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
In [8]: df.tail()
```

```
Out[8]:
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

Figura 3.4: Primeros pasos con Titanic Dataset (parte 4)

```
In [9]: len(df)
```

```
Out[9]: 891
```

```
In [10]: df.shape
```

```
Out[10]: (891, 11)
```

```
In [11]: # faltan algunas columnas dado que son strings.  
df.describe()
```

```
Out[11]:
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Conclusiones:

- Existen columnas con "missing values" (valores que faltan)

Figura 3.5: Primeros pasos con Titanic Dataset (parte 5)

```
In [12]: # y aqui vemos cuantas columnas tienen valores que faltan.
df.isnull().sum()
```

```
Out[12]: Survived      0
Pclass      0
Name        0
Sex         0
Age       177
SibSp       0
Parch       0
Ticket      0
Fare        0
Cabin     687
Embarked     2
dtype: int64
```

```
In [13]: df.Cabin.value_counts()
```

```
Out[13]: G6          4
B96 B98          4
C23 C25 C27      4
F2              3
D              3
..
C99            1
B71            1
A14            1
C103           1
D45            1
Name: Cabin, Length: 147, dtype: int64
```

Figura 3.6: Primeros pasos con Titanic Dataset (parte 6)

```
In [14]: # Los "nan" significa que el dato no fue documentado
for cabina in df.Cabin:
    print(cabina)

nan
C85
nan
C123
nan
nan
E46
nan
nan
nan
G6
C103
nan
nan
nan
nan
nan
nan
```

Figura 3.7: Primeros pasos con Titanic Dataset (parte 7)

```
In [15]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Survived    891 non-null    int64
1   Pclass      891 non-null    int64
2   Name        891 non-null    object
3   Sex         891 non-null    object
4   Age         714 non-null    float64
5   SibSp       891 non-null    int64
6   Parch       891 non-null    int64
7   Ticket      891 non-null    object
8   Fare        891 non-null    float64
9   Cabin       204 non-null    object
10  Embarked    889 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 76.7+ KB
```

Figura 3.8: Primeros pasos con Titanic Dataset (parte 8)

Quiero ver el número aproximado de personas que sobrevivieron

```
In [16]: df.Survived.value_counts()
```

```
Out[16]: 0    549  
         1    342  
         Name: Survived, dtype: int64
```

```
In [17]: df.Survived.value_counts().plot(kind="bar")  
         plt.show()
```

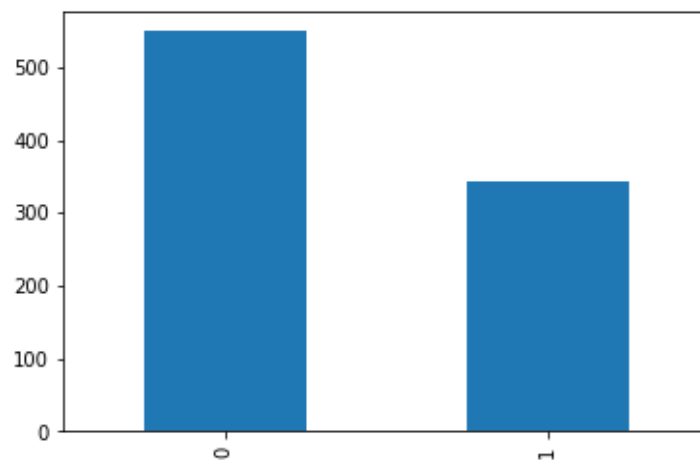


Figura 3.9: Primeros pasos con Titanic Dataset (parte 9)

## 4. SELECCIÓN DE INFORMACIÓN EN DATAFRAMES

¿Cómo seleccionar información concreta de nuestro dataset?

forma 1 de seleccionar información concreta

```
In [18]: df["Age"].head()
Out[18]: 0    22.0
         1    38.0
         2    26.0
         3    35.0
         4    35.0
         Name: Age, dtype: float64
```

forma 2 de seleccionar información concreta

```
In [19]: df.Age.head()
Out[19]: 0    22.0
         1    38.0
         2    26.0
         3    35.0
         4    35.0
         Name: Age, dtype: float64
```

*Figura 4.1: Seleccionando Información de un Dataset (parte 1)*



### Forma 3 de seleccionar información concreta: crosstab

```
In [20]: pd.crosstab(df.Sex, df.Survived)
```

```
Out[20]:
```

	Survived	0	1
Sex			
female		81	233
male		468	109

```
In [21]: pd.crosstab(df.Sex, df.Survived).plot(kind="bar")
plt.show()
```

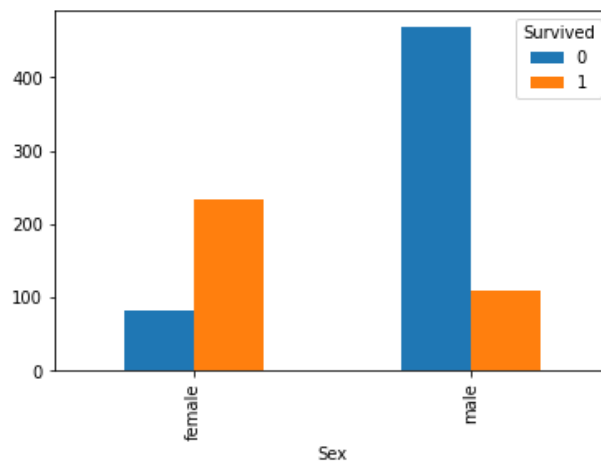


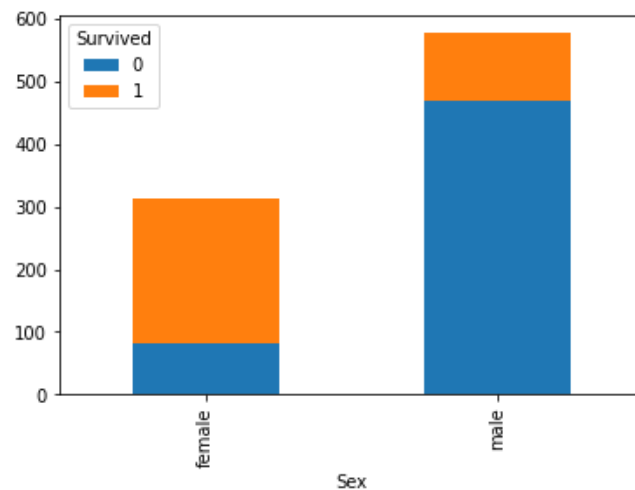
Figura 4.2: Seleccionando Información de un Dataset (parte 2)

#### Conclusiones:

- La mayoría de mujeres sobrevivieron
- La mayoría de hombres no sobrevivieron

Figura 4.3: Seleccionando Información de un Dataset (parte 3)

```
In [22]: # otra forma de visualizarlo  
pd.crosstab(df.Sex, df.Survived).plot(kind="bar", stacked=True)  
plt.show()
```



---

Conclusiones:

- En esta gráfica se puede ver que habían casi el doble de hombres que mujeres
- 

*Figura 4.4: Seleccionando Información de un Dataset (parte 4)*

```
In [23]: pd.crosstab(df.Pclass, df.Survived)
```

```
Out[23]:
```

	Survived	0	1
Pclass			
1	80	136	
2	97	87	
3	372	119	

```
In [24]: pd.crosstab(df.Pclass, df.Survived).plot(kind="bar")
plt.show()
```

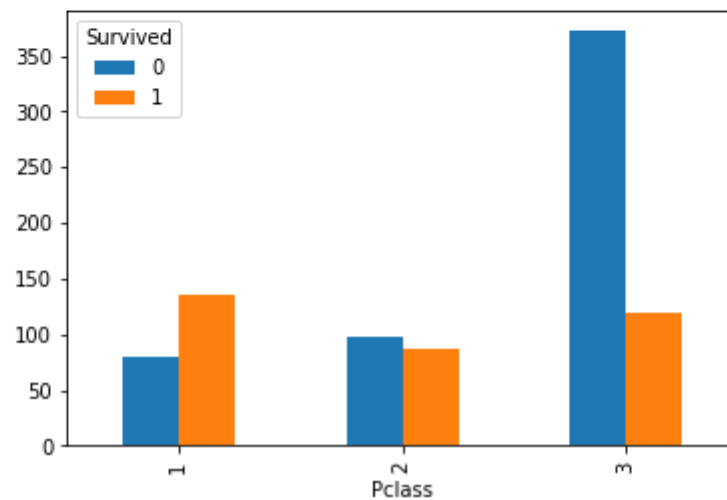


Figura 4.5: Seleccionando Información de un Dataset (parte 5)

#### Conclusiones:

- La mayoría de personas de tercera clase no sobrevivieron

Figura 4.6: Seleccionando Información de un Dataset (parte 6)

## Forma 5 de seleccionar información concreta: groupby

```
In [25]: df.groupby("Sex").Survived.value_counts()
```

```
Out[25]: Sex      Survived
female 1          233
        0           81
male   0          468
        1          109
Name: Survived, dtype: int64
```

```
In [26]: df.groupby("Sex").Survived.value_counts().plot(kind="bar")
plt.show()
```

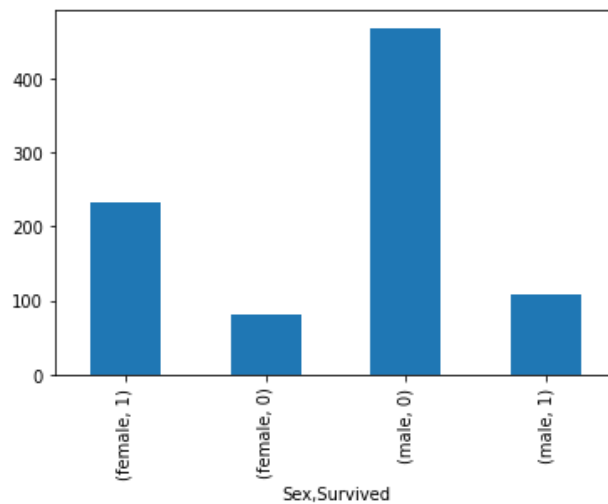


Figura 4.7: Seleccionando Información de un Dataset (parte 7)

## Forma 6 de seleccionar información concreta

Ejemplo:

- Selecciono aquellas filas donde Pclass==1
- Me creo un dataframe de la misma forma que tenía antes

```
In [27]: df_sex_uno = df[df.Pclass==1]
df_sex_uno.head()
```

```
Out[27]:
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
11	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
23	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S

Figura 4.8: Seleccionando Información de un Dataset (parte 8)

```
In [28]: # selecciono
df_sex_uno_crosstab = df[df.Pclass==1]["Survived"]
df_sex_uno_crosstab
```

```
Out[28]: 1      1
        3      1
        6      0
       11      1
       23      1
        ..
      871      1
      872      0
      879      1
      887      1
      889      1
Name: Survived, Length: 216, dtype: int64
```

Figura 4.9: Seleccionando Información de un Dataset (parte 9)

## 5. CREANDO “NUEVOS” DATAFRAMES

### Ejemplos de creación de dataframes concretos

```
In [29]: # supervivencia a 1 --> todos los que sobreviven
df_sobreviven_todos = df[df['Survived']==1]
# supervivencia a 0 --> todos los que no sobreviven
df_sobreviven_ninguno = df[df['Survived']==0]
# supervivencia a 1 - sex = hombre --> Hombres que sobreviven
hombres_supervivientes = df[(df['Survived']==1) & (df['Sex']=="male")]
# supervivencia a 1 - sex = mujer --> Mujeres que sobreviven
mujeres_supervivientes = df[(df['Survived']==1) & (df['Sex']=="female")]
# supervivencia a 0 - sex = hombre --> Hombres que NO sobreviven
hombres_no_supervivientes = df[(df['Survived']==0) & (df['Sex']=="male")]
# supervivencia a 0 - sex = mujer --> Mujeres que NO sobreviven
mujeres_no_supervivientes = df[(df['Survived']==0) & (df['Sex']=="female")]
```

Figura 5.1: Creando nuevos DataFrames (parte 1)

```
In [30]: # probamos..
```

```
In [31]: df_sobreviven_todos.head(3)
```

```
Out[31]:
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

```
In [32]: df_sobreviven_todos.Survived.value_counts(3)
```

```
Out[32]: 1    1.0
Name: Survived, dtype: float64
```

Figura 5.2: Creando nuevos DataFrames (parte 2)

In [33]: df\_sobreviven\_ninguno.head(3)

Out[33]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

In [34]: hombres\_supervivientes.head(3)

Out[34]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
17	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0	NaN	S
21	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0	D56	S
23	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5	A6	S

Figura 5.3: Creando nuevos DataFrames (parte 3)

In [35]: mujeres\_supervivientes.head(3)

Out[35]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

In [36]: hombres\_no\_supervivientes.head()

Out[36]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S

Figura 5.4: Creando nuevos DataFrames (parte 4)

In [37]: mujeres\_no\_supervivientes.head()

Out[37]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
14	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8542	NaN	S
18	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763	18.0000	NaN	S
24	0	3	Palsson, Miss. Torborg Danira	female	8.0	3	1	349909	21.0750	NaN	S
38	0	3	Vander Planke, Miss. Augusta Maria	female	18.0	2	0	345764	18.0000	NaN	S
40	0	3	Ahlin, Mrs. Johan (Johanna Persdotter Larsson)	female	40.0	1	0	7546	9.4750	NaN	S

Figura 5.5: Creando nuevos DataFrames (parte 5)

## 6. OBTENCIÓN DE INFORMACIÓN DE LOS GRÁFICOS

### Obtenemos información de los gráficos

#### Función para hacer gráficas de forma automática

In [38]: `df.head(2)`

Out[38]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C

Figura 6.1: Obtención de Información de los Gráficos (parte 1)

```
In [39]: # pd.crosstab(df["Sex"], df.Survived)

opciones = ["Pclass", "Sex", "Embarked"]

for opcion in opciones:
    pd.crosstab(df[opcion], df.Survived).plot(kind="bar")
    plt.show()
```

Figura 6.2: Obtención de Información de los Gráficos (parte 2)



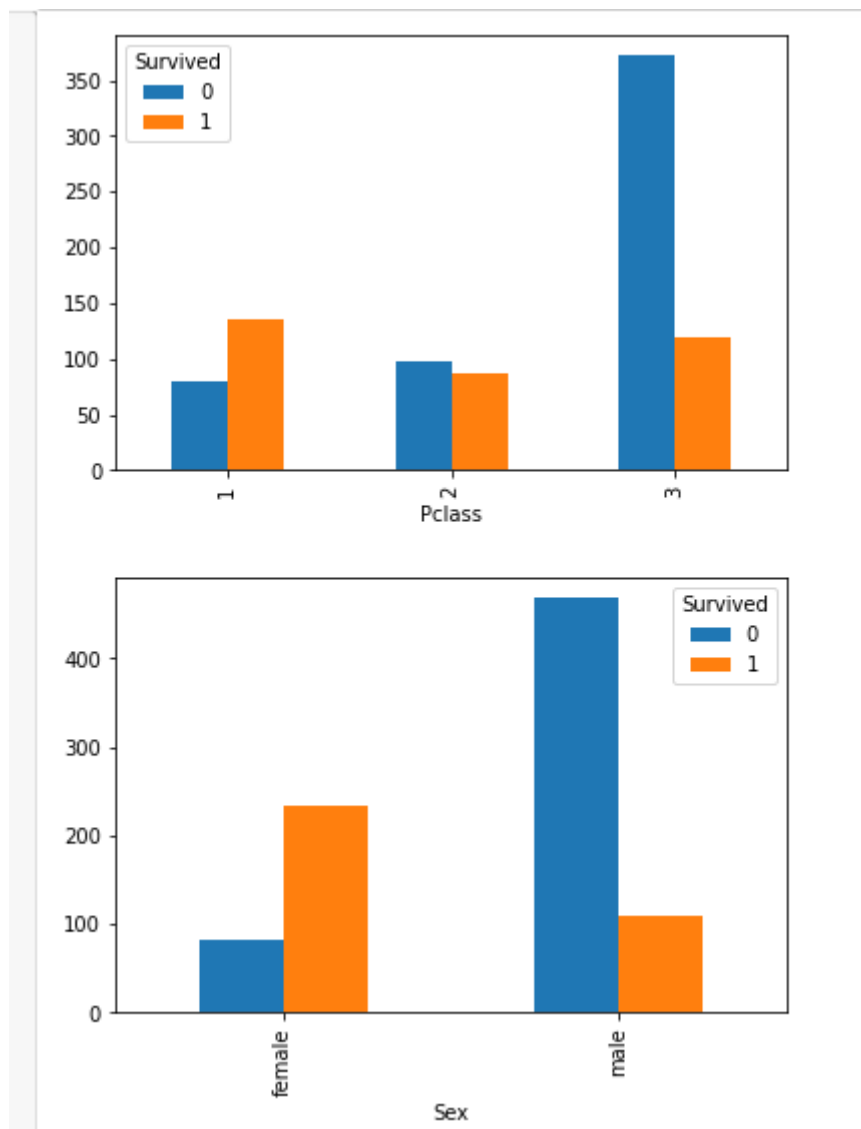


Figura 6.3: Obtención de Información de los Gráficos (parte 3)

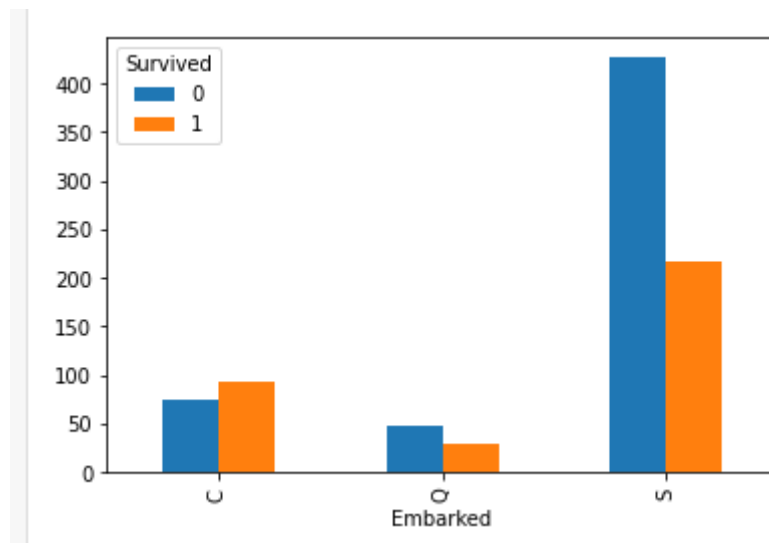


Figura 6.4: Obtención de Información de los Gráficos (parte 4)

#### Algunos Gráficos de Seaborn

```
In [40]: # UserWarning: The `factorplot` function has been renamed to `catplot`.
# The original name will be removed in a future release
sns.factorplot('Sex', 'Survived', hue='Pclass', size=4, aspect=2, legend=True, data=df)
plt.show()

c:\users\manut\appdata\local\programs\python\python38\lib\site-packages\seaborn\categorical.py:3714: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` ('point') has changed to `strip` in `catplot`.
  warnings.warn(msg)
c:\users\manut\appdata\local\programs\python\python38\lib\site-packages\seaborn\categorical.py:3720: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
c:\users\manut\appdata\local\programs\python\python38\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Figura 6.5: Obtención de Información de los Gráficos (parte 5)

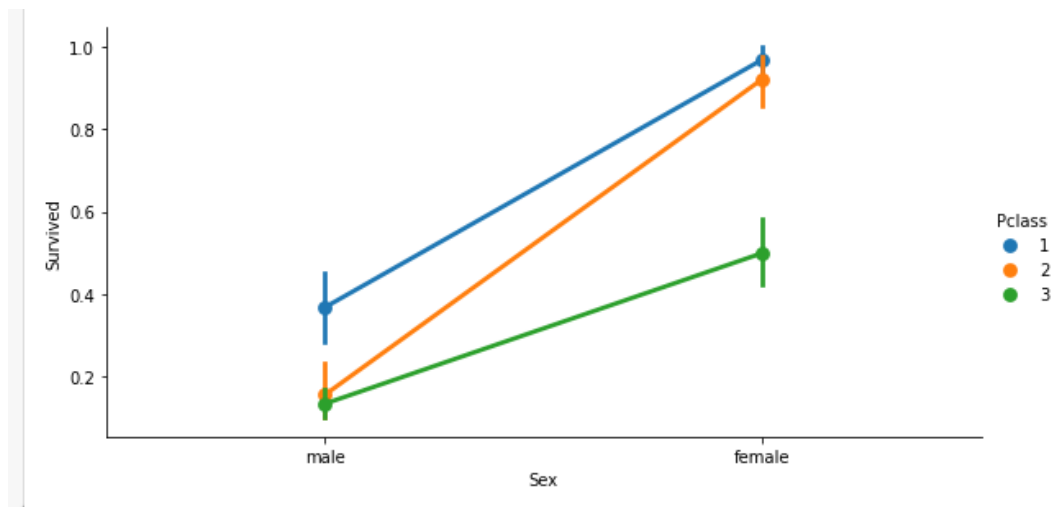


Figura 6.6: Obtención de Información de los Gráficos (parte 6)

```
In [41]: # https://seaborn.pydata.org/generated/seaborn.catplot.html
sns.catplot(x='Sex', y='Survived', hue='Pclass', kind="strip", data=df)
plt.show()
```

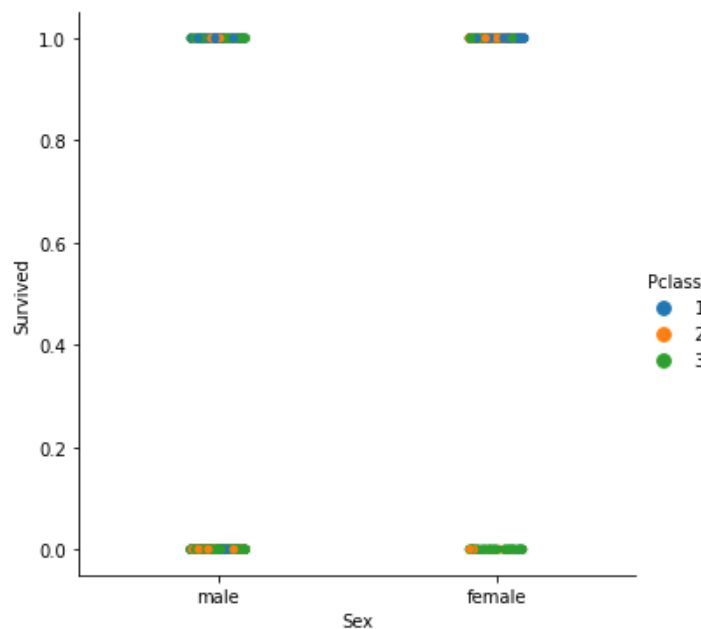


Figura 6.7: Obtención de Información de los Gráficos (parte 7)

```
In [42]: sns.catplot(x='Sex', y='Survived', hue='Pclass', kind="point", height=4, aspect=2, data=df)
plt.show()
```

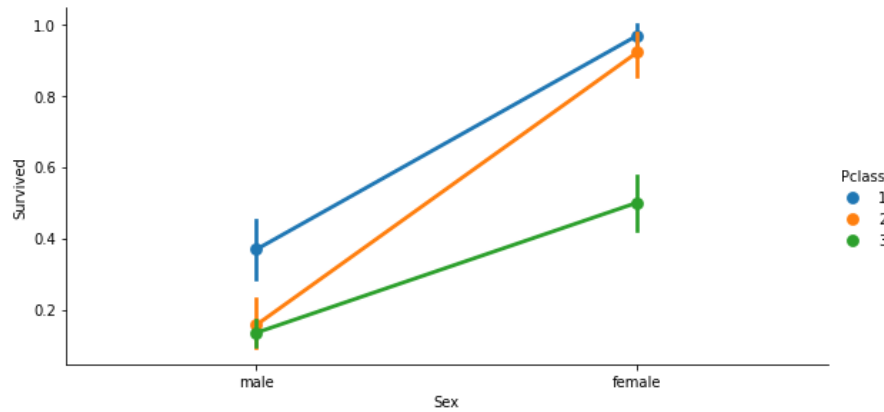


Figura 6.8: Obtención de Información de los Gráficos (parte 8)

```
In [43]: sns.catplot(x='Pclass', y='Survived', hue='Sex', col='Embarked', kind="point", data=df)
plt.show()
```

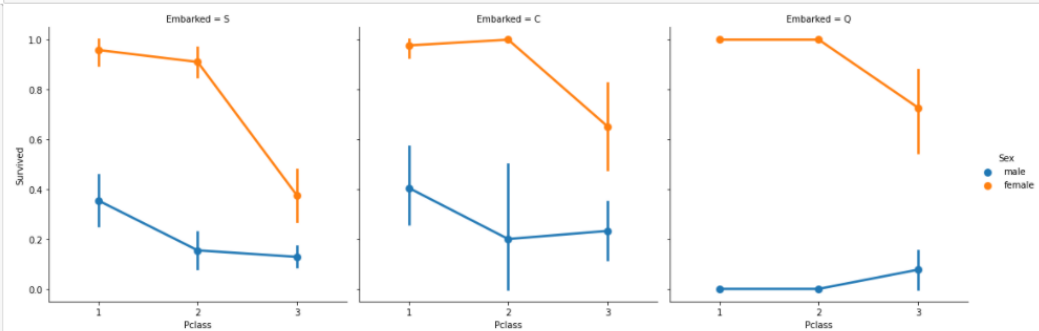


Figura 6.9: Obtención de Información de los Gráficos (parte 9)

Algunas conclusiones:

- Nos fijamos en La gráfica de la izquierda, embarked="S"  
Las **mujeres** de **3 clase** que embarcaron en **S**  
fallecieron muchas en comparacion con 1 y 2 clase.  
pese a ello sobrevivieron algo mas que lo hombres de 1 clase embarcando del mismo puerto.
- Los **hombres** con mayor porcentaje de **supervivencia** embarcaron en **C**
- Los hombres con menor porcentaje de supervivencia embarcaron en Q
- Vemos nuevamente como la mayoría de mujeres sobrevivió, pero no los hombres.

Figura 6.10: Obtención de Información de los Gráficos (parte 10)

## Edad y Supervivencia

```
In [44]: # me creo una figura
fig = plt.figure(figsize=(16,6))
# 3 subplots
# 1 fila 3 columnas - gráfica 1
ax1 = fig.add_subplot(131)
# 1 fila 3 columnas - gráfica 2
ax2 = fig.add_subplot(132)
# 1 fila 3 columnas - gráfica 3
ax3 = fig.add_subplot(133)

# violinplot
sns.violinplot(x="Embarked", y="Age", hue="Survived", data=df, ax=ax1)
sns.violinplot(x="Pclass", y="Age", hue="Survived", data=df, ax=ax2)
sns.violinplot(x="Sex", y="Age", hue="Survived", data=df, ax=ax3)

plt.show()
```

Figura 6.11: Obtención de Información de los Gráficos (parte 11)

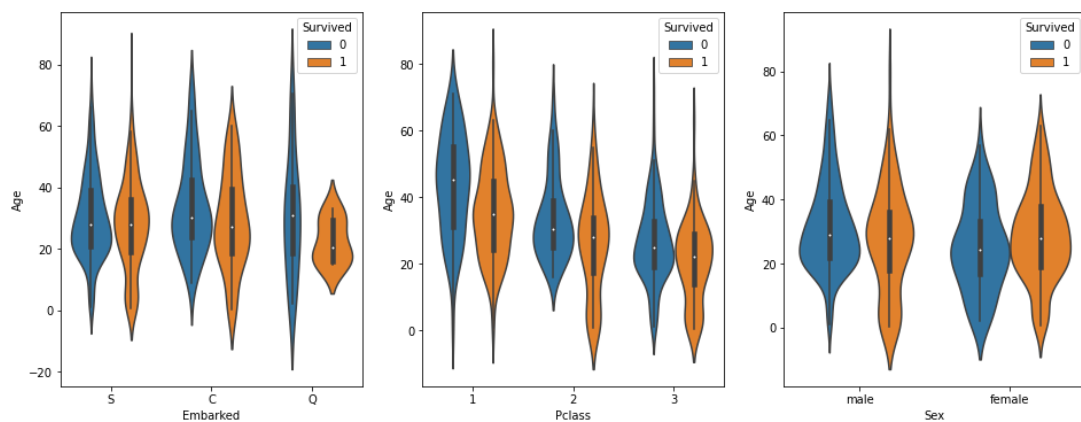


Figura 6.12: Obtención de Información de los Gráficos (parte 12)

Hago un `split=True` para que me lo haga más visual

```
In [45]: # me creo una figura
fig = plt.figure(figsize=(16,6))
# 3 subplots
# 1 fila 3 columnas - gráfica 1
ax1 = fig.add_subplot(131)
# 1 fila 3 columnas - gráfica 2
ax2 = fig.add_subplot(132)
# 1 fila 3 columnas - gráfica 3
ax3 = fig.add_subplot(133)

# violinplot
sns.violinplot(x="Embarked", y="Age", hue="Survived", data=df, split=True, ax=ax1)
sns.violinplot(x="Pclass", y="Age", hue="Survived", data=df, split=True, ax=ax2)
sns.violinplot(x="Sex", y="Age", hue="Survived", data=df, split=True, ax=ax3)

plt.show()
```

Figura 6.13: Obtención de Información de los Gráficos (parte 13)

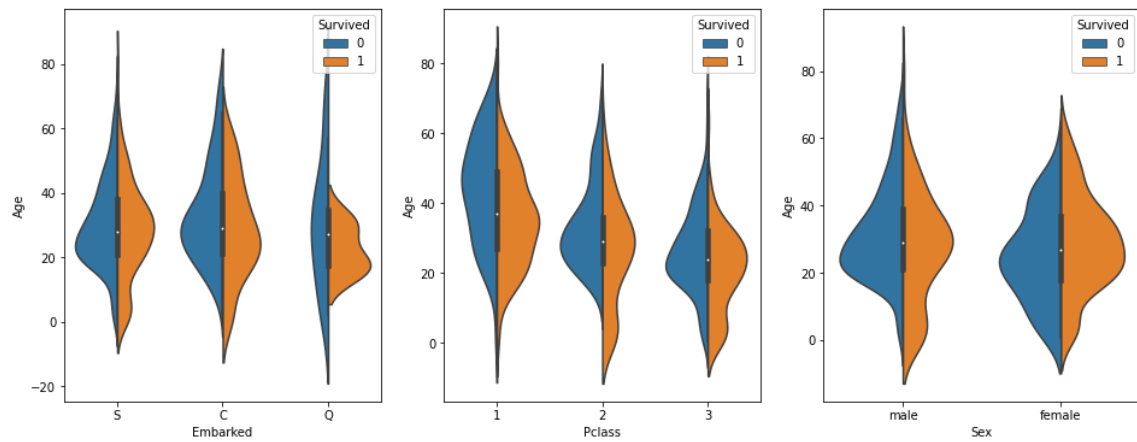


Figura 6.14: Obtención de Información de los Gráficos (parte 14)

#### Conclusiones:

- EMBARKED y Age:
  - La gente de unos 18-35 años de Q SI sobrevivieron mayoritariamente,(no todos)
  - no hay porcentajes mayoritarios significativos en las otras 2 embarcaciones
  - En Q embarcaron bastantes niños los cuales no sobrevivieron.
- PCLASS y Age:
  - De la 2ª clase sobre todo y la 3 sobrevivieron la mayoría de sus niños
- Sex y Age:
  - Hay mas ancianos que ancianas
  - Los jovenes (varón) menores de 20 años en general sobrevivieron pero no las mujeres

Figura 6.15: Obtención de Información de los Gráficos (parte 15)

```
In [46]: df.Age.describe()
# min = 0.42
# max = 80
```

```
Out[46]: count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

Figura 6.16: Obtención de Información de los Gráficos (parte 16)



## heatmap

```
In [47]: plt.figure(figsize=(8,8))
sns.heatmap(df.corr(), annot=True)
plt.show()
```

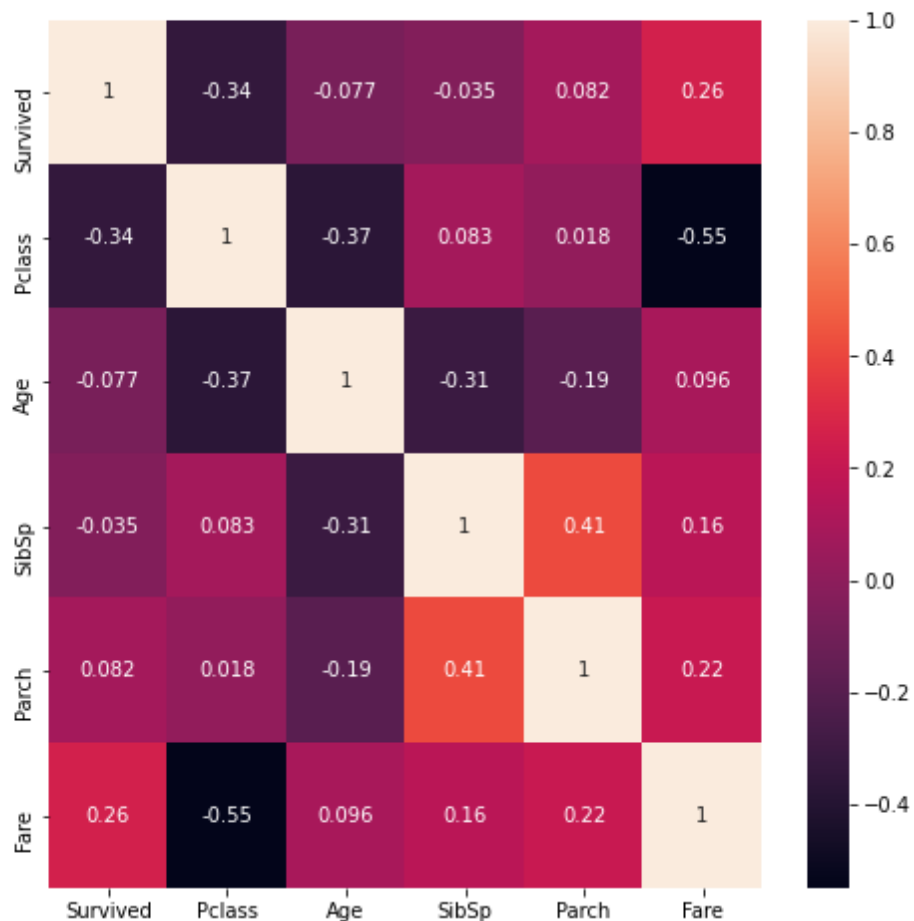


Figura 6.17: Obtención de Información de los Gráficos (parte 17)

## barplot

```
In [48]: sns.barplot(x="Pclass", y="Survived", data=df)
plt.show()
```

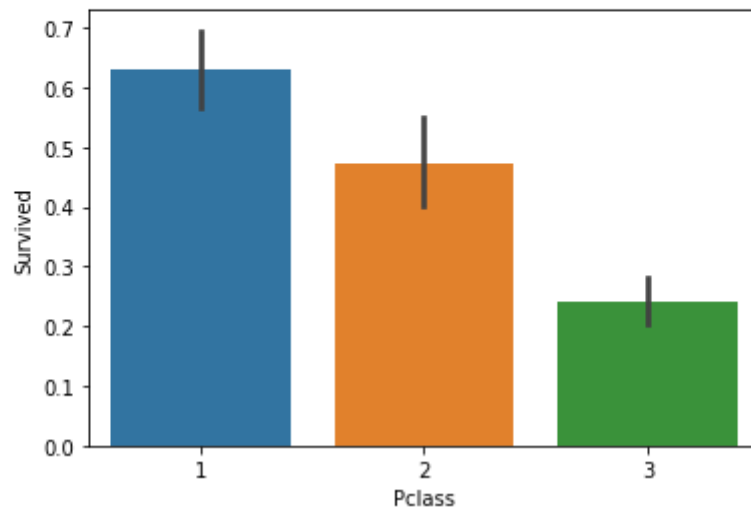


Figura 6.18: Obtención de Información de los Gráficos (parte 18)

```
In [49]: def funcion_graficas(feet):
plt.subplot(2,1,1)
df.groupby(feet).Survived.value_counts().plot(kind="bar")
plt.figure(figsize=(12,8))
plt.subplot(2,1,2)
sns.barplot(x=feet, y="Survived", data=df)
plt.show()
```

Figura 6.19: Obtención de Información de los Gráficos (parte 19)

```
In [50]: funcion_graficas("Pclass")
```

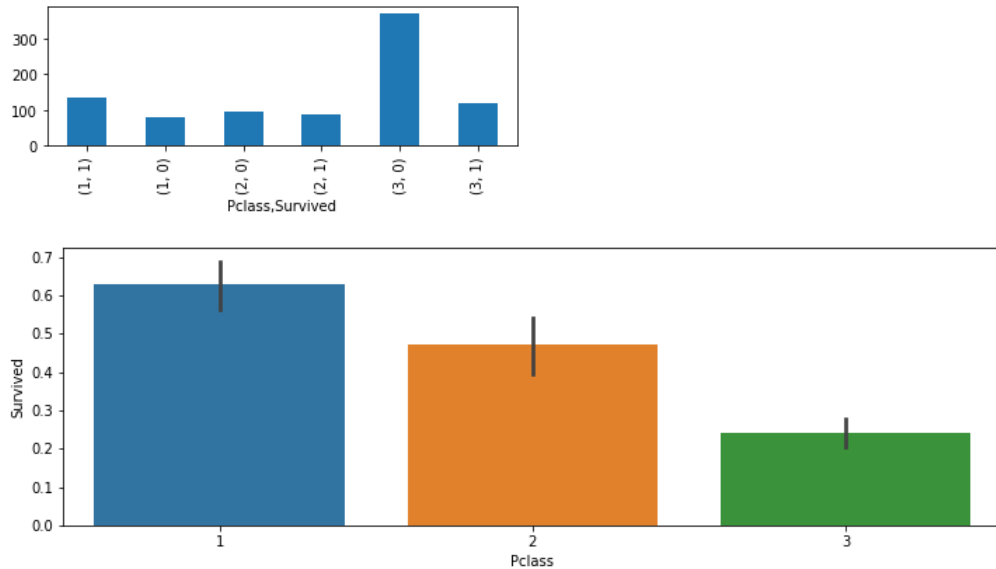


Figura 6.20: Obtención de Información de los Gráficos (parte 20)

```
In [51]: funcion_graficas("Sex")
```

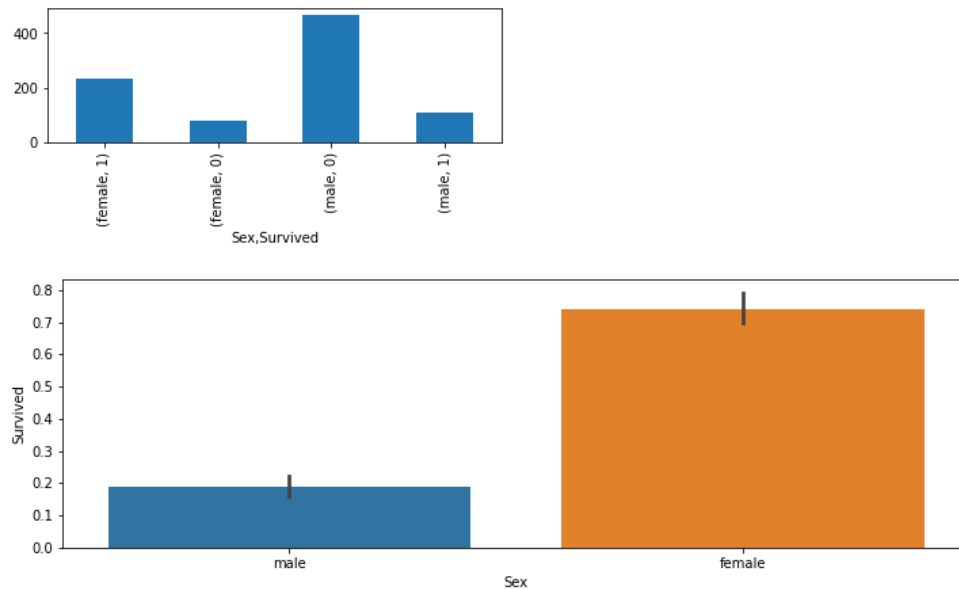


Figura 6.21: Obtención de Información de los Gráficos (parte 21)

```
In [52]: funcion_graficas("Age")
```

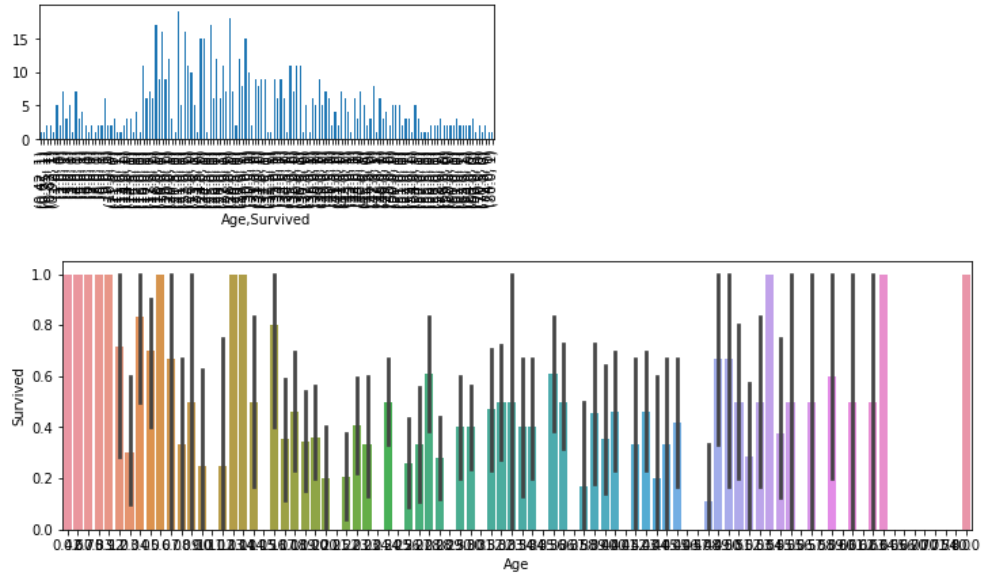


Figura 6.22: Obtención de Información de los Gráficos (parte 22)

```
In [53]: funcion_graficas("SibSp")
```

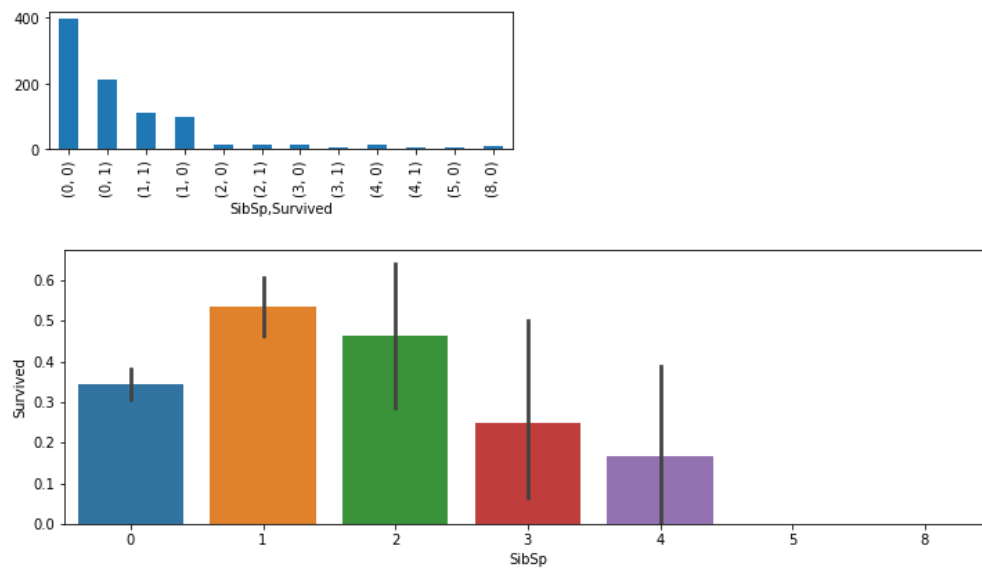


Figura 6.23: Obtención de Información de los Gráficos (parte 23)

In [54]: `funcion_graficas("Parch")`

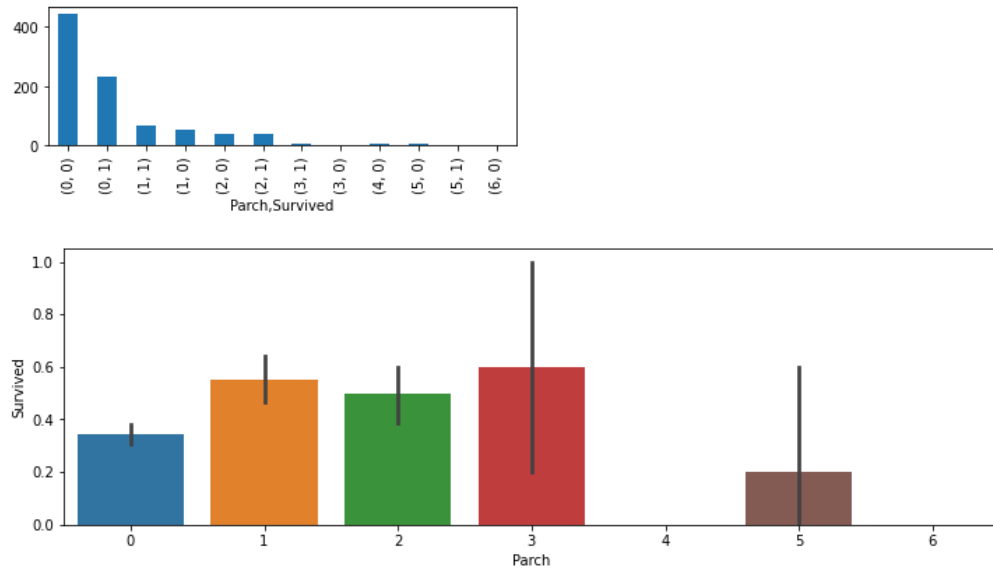


Figura 6.24: Obtención de Información de los Gráficos (parte 24)

In [55]: `funcion_graficas("Embarked")`

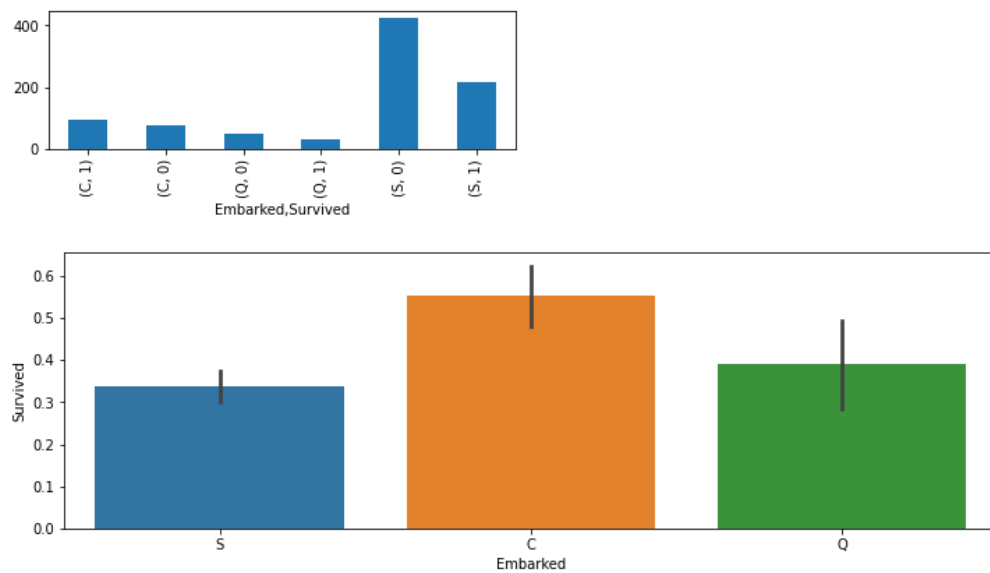


Figura 6.25: Obtención de Información de los Gráficos (parte 25)

```
In [56]: # Ahora se podrían sacar algunas conclusiones más  
# por el momento es suficiente!
```

Figura 6.26: Obtención de Información de los Gráficos (parte 26)

## 7. PUNTOS CLAVE

- | Existen muchas opciones para aprender Data Science, siendo Kaggle una de ellas.
- | El Dataset del Titanic es una buena elección para aprender a hacer Gráficas de muchos tipos.

